Remarks

Entrance of this amendment and reconsideration of the pending claims are respectfully requested. Upon entrance of this amendment, claims 12-19 will remain pending.

Initially, Applicants address the 35 U.S.C. §112, first paragraph rejections to claims 12-20 by deleting the phrase "in hardware" from claim 12. Notwithstanding this amendment, Applicants respectfully submit that one of ordinary skill in the art reading Applicants' specification would understand that the dedicated collective offload engine provides collective processing in hardware of data from the at least some processing nodes. The dedicated collective offload engine is expressly recited in claim 12 to be a hardware device coupled to the switch fabric. This hardware device, described for example, in Applicants' specification paragraph [0015], provides the collective processing of data from the at least some processing nodes. One example of the dedicated collective offload engine is depicted in Applicants' FIG. 2. The components of dedicated collective offload engine 200 in FIG. 2 are hardware components. There is no provision in the circuitry depicted for, nor need for, a processor employing software code. As such, Applicants' dedicated collective offload engine provides collective processing in hardware of data from the at least some processing nodes.

Applicants' dedicated collective offload engine is further recited in claim 12 to include a dispatcher built from field programmable gate arrays, and a pipelined arithmetic logic unit. Both of these hardware components are depicted in FIG. 2 and described, for example, in specification paragraph [0017]. The dispatcher controls collective processing of data by the arithmetic logic unit.

Based upon the above-noted amendments, Applicants respectfully request reconsideration and withdrawal of the 35 U.S.C. §112, first paragraph, rejections to prior pending claims 12-20.

Prior claims 12, 14-17 & 20 were rejected under 35 U.S.C. §102(e) as being anticipated by Burianek et al. (U.S. Patent No. 7,082,457; hereinafter Burianek), and claims 13, 18 & 19 were rejected under 35 U.S.C. §103(a) as being unpatentable over Burianek. These rejections are respectfully traversed, and reconsideration thereof is requested.

Applicants' claim 12 recites providing, by a *dedicated collective offload engine* coupled to a switch fabric in a distributed parallel computing system, collective processing of data. There is no dedicated collective offload engine in Burianek as the term is employed in Applicants' specification and claims. Further, there is no collective processing of data in Burianek. In the final Office Action, Applicants' recited dedicated collective offload engine is analogized to server 215 of Burianek. This analogy is respectfully traversed.

Server 215 in Burianek is described as a project management central server that directs signals sent to and from the components of the distributed computing environment. This server includes a delegation component which sends and receives information about project tasks stored in the database 210. Thus, server 215 in Burianek is a conventional server system. This is distinguished from Applicants' *dedicated collective offload engine*, which provides collective processing of data. In Applicants' invention, the *dedicated collective offload engine is a hardware device* coupled to the switch fabric. This hardware device is believed to distinguish Applicants' invention from Burianek. In Burianek, the processing described is implemented in software, while in Applicants' recited invention, collective processing is implemented in a hardware device, that is, the dedicated collective offload engine.

With respect to the dedicated collective offload engine being implemented *as a hardware device*, the final Office Action references column 4, lines 20-45, wherein Burianek teaches that the server is a computer, which is a hardware device with software. Responsive to this characterization, Applicants submit amendments to claim 12 herewith which further define the dedicated offload engine as a hardware device. Specifically, Applicants recite that the dedicated collective offload engine is a hardware device which comprises a dispatcher built from field programmable gate arrays, and a pipelined arithmetic logic unit. The dispatcher controls collective processing of data by the arithmetic processing unit. Further, Applicants recite that the collective processing *implements a collective operation* on data from the at least some processing nodes *without use of a software tree*. In prior collective processing approaches, a software tree is typically employed in implementing a collective operation.

Applicants recite a data processing method which includes collective processing of data from the at least some processing nodes of the multiple processing nodes of a distributed, parallel computing system. The collective processing implements a collective operation on the data from the at least some processing nodes. The phrases "*collective processing*" and "*collective operation*" are terms of art which *refer to a particular type of data processing*. A collective operation is conventionally an arithmetic operation executed across data from multiple nodes of a distributed, parallel computing system, with results being provided to multiple nodes. Thus, a collective operation is an n:n operation.

As explained in Applicants' "Background of the Invention", implementation of collective processing typically includes using a software tree approach, wherein message passing facilities are used to form a virtual tree of processes. A drawback to this approach is the serialization of delays at each stage of the tree. These delays are additive in the overall overhead associated with the collective processing. Furthermore, the software tree approach results in a theoretical logarithmic scaling latency of the overall collective processing versus system size. Due to interference from daemons, interrupts and other background activity, cross traffic, and the unsynchronized nature of independent operating system images and their dispatch cycles, measured values of scaling latency are usually significantly worse than theoretical values. Responsive to this issue, Applicants describe a novel *collective processing approach* with mitigates the large latency associated with the software tree implementation. In Applicants' approach, a dedicated collective offload engine, which is a hardware device coupled to the switch fabric, is employed to provide the collective processing of data from the multiple processing nodes. Applicants' hardware device is a specialized device dedicated to providing the collective processing in hardware of the data, and the collective processing implements a collective operation on the data. The collective processing occurs in hardware since the device itself is a hardware device, and no software tree (or software program) is employed to perform the collective operation.

An Internet search on the phrase "collective operation" in a distributed parallel computing system, or "collective processing" provides support for the above-noted meaning of these phrases, as employed in the art. Applicants respectfully request that this meaning be given

consideration when evaluating the claims at issue. Burianek does not describe collective processing *per se*, nor is a collective operation, as the term is understood in the art, described in Burianek.

For at least the above-noted reasons, Applicants respectfully request reconsideration and allowance of the independent claim presented herewith. Applicants specifically recite that the dedicated collective offload engine, which is a hardware device coupled to the switch fabric, is a specialized device dedicated to providing collective processing of data from at least some processing nodes of a distributed, parallel computing system. *The dedicated collective offload engine includes a dispatcher built from field programmable gate arrays and a pipelined arithmetic logic unit.* The dispatcher controls the collective processing of data by the arithmetic logic unit, and the collective processing implements a collective operation on the data without use of a software tree.
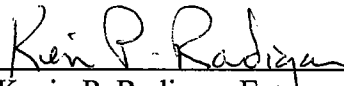
The dependent claims are believed allowable for the same reasons as the independent claim, as well as for their own additional characterizations.

For example, claim 13 recites that the collective operation is a message passing interface (MPI) operation. An MPI collective operation is a particular type of collective operation implemented within an MPI standard. Details on MPI collective operations are provided at http://www.redbooks.ibm.com/redbooks/pdfs/sg245380.pdf. For example, reference Chapter 2 thereof. There is no discussion in Burianek of the MPI standard, or of a collective operation implemented within the standard. Thus, there is no discussion in Burianek of an MPI collective operation being implemented by a dedicated collective offload engine which is a hardware device coupled to the switch fabric and which comprises a dispatcher, built from field programmable gate arrays, and a pipelined arithmetic logic unit. No such device is taught or suggested by the art of record.

All claims are believed to be in condition for allowance, and such action is respectfully requested.

Should the Examiner have reservations regarding the patentability of any claim(s) presented, Applicants' undersigned representative respectfully requests the opportunity for an Examiner Interview to discuss the claim(s) in the hope of advancing prosecution of the subject application.

Respectfully submitted,

Kevin P. Radigan, Esq.
Attorney for Applicants
Registration No.: 31,789

Dated: August 29, 2008.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579